

**APLIKASI ALGORITMA GENETIKA MULTI OBYEKTIF PADA  
*TRAVELING SALESMAN PROBLEM***

**(Studi Kasus: Mini Bus Rute Sukajadi-Panam Pekanbaru)**

**TUGAS AKHIR**

Diajukan sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Sains Pada  
Jurusan Matematika

Oleh:

**RONI ANSHARY**

**10354023067**



**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU  
PEKANBARU  
2010**

# **APLIKASI ALGORITMA GENETIKA MULTI OBYEKTIF PADA *TRAVELING SALESMAN PROBLEM***

**(Studi kasus: mini bus rute sukajadi-panam pekanbaru)**

**RONI ANSHARY**

**10354023067**

Tanggal Sidang : 08 Februari 2010

Periode Wisuda : Juli 2010

Jurusan Matematika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

## **ABSTRAK**

Penelitian ini membahas tentang *Travelling Salesman Problem* dengan studi kasus pada *mini bus* sukajadi-panam untuk menentukan rute terpendek dengan menggunakan *multi obyektif genetika algorithms* (MOGA). Penyelesaian algoritma genetika tersebut menerapkan pemograman *Borland Delphi 7.0*. berdasarkan hasil penelitian diperoleh bahwa lintasan dengan jarak terpendek pada kromosom I dengan jarak 7800 Meter. Lintasan dengan waktu terpendek didapatkan pada kromosom III dengan waktu 165,43 Menit. Sedangkan lintasan dengan biaya paling sedikit didapatkan pada kromosom I dengan biaya Rp12500,00.

**Kata kunci:** Algoritma Genetika Multi Objective, Borland Delphi 7.0, Traveling Salesman Problem

# **APLIKASI ALGORITMA GENETIKA MULTI OBYEKTIF TO TRAVELING SALESMAN PROBLEM**

**(Studi Kasus: Mini Bus Rute Sukajadi-Panam Pekanbaru)**

**RONI ANSHARY**

**NIM: 10354023067**

*Date of Final Exam : 08 February 2010*

*Graduation Ceremony Priode : July 2010*

*Mathematic Departement*

*Faculty of Sciences and Technology*

*State Islamic University of Sultan Syarif Kasim Riau*

*Jl. HR. Soebrantas No 155 Pekanbaru*

## **ABSTRACT**

*This research study about Travelling of Salesman Problem with the case study at mini bus Sukajadi-Panam to determine the short route by using objective multi genetika algoritms (MOGA). Solving of the algorithm genetika of menerapkan pemograman Borland Delphi 7.0. pursuant to research result obtained that trajectory with the short distance at chromosome I with the distance 7800 Metre. Trajectory with the short time got chromosome III with the time 165,43 Minute. While trajectory with the least expense got chromosome I with the expense Rp12500,00.*

**Keyword:** *Borland Delphi 7.0, Multi Objective Genetica Algoritms, Travelling of Salesman Problem*

## DAFTAR ISI

Halaman

LEMBAR PERSETUJUAN .....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL .....	iv
LEMBAR PERNYATAAN .....	v
ABSTRAK .....	vi
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiii
DAFTAR LAMPIRAN .....	xiii
 BAB I. PENDAHULUAN.....	 I-1
1.1 Latar Belakang .....	I-1
1.2 Perumusan Masalah .....	I-2
1.3 Tujuan Pulisan .....	I-2
1.4 Sistematika Penulisan .....	I-2
BAB II. LANDASAN TEORI.....	II-1
2.1 Kromosom.....	II-1
2.1.1 Macam-macam Bentuk Kromosom .....	II-2
2.2 Fungsi <i>Fitness</i> .....	II-2
2.2.1 Starategi Menentukan Fungsi <i>Fitness</i> .....	II-2
2.3 Bilangan <i>Biner</i> .....	II-3
2.4 persilangan ( <i>crossover</i> ) .....	II-4
2.4.1 Starategi Penentuan <i>Crossover</i> .....	II-5
2.5 Mutasi .....	II-6
2.5.1 Starategi Menentukan Mutasi .....	II-7
2.6 Algoritma Genetika.....	II-7
BAB III. METODOLOGI PENELITIAN .....	III-1
BAB IV. APLIKASI ALGORITMA GENETIKA <i>MULTI OBYEKTIF</i> PADA	
<i>TRAVELLING SALESMAN PROBLEM</i> .....	IV-1
4.1 <i>Travelling Slesman Problem</i> (TSP) .....	IV-1
4.2 TSP <i>Multi Obyektif</i> .....	IV-2
4.2.1 Bahasa Pemograman yang digunakan .....	IV-3

4.3 Tampilan Aplikasi .....	IV-3
4.4 Contoh Aplikasi .....	IV-4
BAB V. KESIMPULAN DAN SARAN .....	V-1
5.1 Kesimpulan .....	V-1
5.2 Saran .....	V-1
LAMPIRAN	
DAFTAR PUSTAKA	
DAFTAR RIWAYAT HIDUP	

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring perkembangan struktur data dan algoritma yang semakin baik, maka banyak persoalan manusia dalam kehidupan sehari-hari yang mampu diselesaikan dengan perhitungan komputasi. Namun selama ini kebanyakan masalah yang diselesaikan adalah terbatas pada sebuah varian saja dan bersifat linier. Padahal fakta dilapangan pasti tidak dapat dilihat dari satu aspek saja, dengan kata lain terdiri dari banyak aspek dan bersifat *Nondeterministic Polynomial-time Complete* (NPC) artinya parameter yang mempengaruhi sangat kompleks dan penyelesaiannya tidak dapat diselesaikan dengan sebuah algoritma linier saja, karena membutuhkan waktu yang sangat lama. Seperti halnya masalah Traveling Salesman Problem yang mana jumlah kemungkinan penyelesaian semakin meningkat seiring bertambahnya jumlah kota.

Namun selama ini permasalahan TSP masih hanya sebatas mencari penyelesaian dengan jarak terpendek, yang belum sepenuhnya bisa diaplikasikan dalam persoalan di kehidupan kita, karena itu banyak perkiraan yang mestinya dimasukkan misalnya biaya(*cost*), kenyamanan, keamanan, kemacetan, sehingga benar-benar merupakan bentuk permasalahan dalam kehidupan manusia dan membantu manusia jika diperoleh suatu penyelesaiannya.

Algoritma genetika digunakan untuk menyelesaikan permasalahan searching dan optimasi yang mempunyai kompleksitas tinggi yang banyak terjadi dalam dynamic programming seperti TSP dan Knapsack Problem. Algoritma genetika dapat menghindari keadaan lokal optimum yang baik. Pengembangan model parallel computing dapat dengan mudah dilakukan.

*Traveling Salesman Problem* (TSP) merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti *routing* dan penjadwalan produksi. Masalah optimasi TSP terkenal dan telah menjadi standar untuk mencoba algoritma yang komputasional. Pokok permasalahan dari TSP

adalah seorang *salesman* harus mengunjungi sejumlah kota yang diketahui jaraknya satu dengan yang lainnya. Semua kota yang ada harus dikunjungi oleh *salesman* tersebut dan kota tersebut hanya boleh dikunjungi tepat satu kali. Permasalahannya adalah bagaimana *salesman* tersebut dapat mengatur rute perjalanannya sehingga jarak yang ditempuhnya merupakan jarak minimum.

Penelitian ini pendekatan yang digunakan adalah algoritma genetik. Algoritma genetik ditemukan oleh John Holland pada tahun 1960. Algoritma ini menerapkan suatu proses evolusi biologi. Banyak percobaan dalam menyelesaikan *traveling salesman problem* dengan menggunakan metode algoritma genetik, tetapi masih terdapat beberapa percobaan yang menghasilkan solusi yang tidak layak. Pada metode dalam penelitian ini, ketidak layakan dari solusi dapat dihindari dengan menggunakan suatu skema yang menjelaskan permasalahan kasus TSP dengan baik dan lebih terstruktur.

Berdasarkan latar belakang di atas, maka saya tertarik melakukan penelitian dengan judul “aplikasi algoritma genetika multi obyektif pada *traveling salesman problem* ( studi kasus: mini bus rute sukajadi-panam pekanbaru).

## **1.2 Perumusan Masalah**

Perumusan masalah dalam penelitian ini adalah bagaimana cara menentukan rute terpendek dalam *travelling salesman problem* dengan menggunakan *multi objective genetic algorithms*.

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah Menerapkan *algoritma genetika multi obyektif* untuk mendapatkan rute terpendek dalam *traveling salesman problem*.

## **1.4 Sistematika Penulisan**

Sistematika dalam pembuatan tulisan ini mencakup lima bab yaitu :

### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan masalah, dan sistematika penulisan.

## **Bab II      Landasan Teori**

Bab ini berisi informasi tentang teori-teori yang digunakan dalam penulisan ataupun metode yang dipakai.

## **Bab III    Metodologi Penelitian**

Bab ini berisikan langkah-langkah dalam menyelesaikan *travelling salesman problem* dengan menggunakan *algoritma genetika multi obyektif*.

## **Bab IV    Pembahasan dan Analisa**

Bab ini berisikan penyelesaian *travelling salesman problem* dengan menggunakan *algoritma genetika multi obyektif*.

## **Bab V      Penutup**

Bab ini berisikan kesimpulan dan saran.

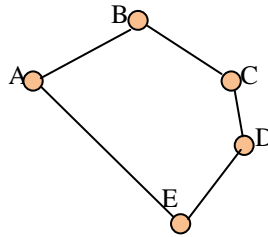


## BAB II

### LANDASAN TEORI

#### 2.1 Kromosom

Representasi pengkodean yang digunakan pada algoritma genetika adalah representasi kunci-kunci random. Representasi ini mengkodekan sebuah gen dengan membangkitkan bilangan acak antara (0,1). Nilai acak dalam posisi  $i$  menentukan urutan didatanginya kota  $i$  dalam lintasan TSP. Dapat dilihat contoh dari lintasan TSP pada gambar berikut:



Gambar 2.1. lintasan TSP

Keterangan:

Kota A	= Titik awal
Kota B	= 2 km
Kota C	= 3 km
Kota D	= 4 km
Kota E	= 6 km

Posisi  $i$  dalam *list* menunjukkan kota  $i$ . Nilai acak dalam posisi  $i$  menentukan urutan didatanginya kota  $i$  dalam lintasan TSP. Titik awal dalam lintasan TSP ini kota A, sehingga kota A menempati urutan pertama, kemudian kota B merupakan jarak terdekat dari kota A, jadi kota B berada pada urutan kedua, sehingga urutannya menjadi:

$A - B - C - D - E$

### 2.1.1 Macam-macam bentuk kromosom

Macam-macam bentuk kromosom yaitu:

- a. Kromosom *biner*, yaitu kromosom yang disusun dari gen-gen yang bernilai 0 dan 1. Kromosom ini adalah model standar dalam algoritma genetika.
- b. Kromosom *float*, yaitu kromosom yang disusun dari gen-gen dengan nilai pecahan. Gen integer dapat digolongkan dalam bentuk kromosom ini.
- c. Kromosom *string*, yaitu kromosom yang disusun dari gen-gen yang bernilai string (simbol).
- d. Kromosom *kombinatorial*, yaitu kromosom yang disusun dari gen-gen yang dinilai dari urutannya.

## 2.2 Fungsi *fitness*

Fungsi *fitness* digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan.

### 2.2.1 Strategi menentukan nilai fungsi *fitness*

Strategi dalam menentukan nilai fungsi *fitness* dapat dilakukan dengan cara:

- a. Nilai *fitness* merupakan suatu ukuran baik tidaknya suatu solusi yang dinyatakan sebagai satu individu, atau dengan kata lain nilai *fitness* menyatakan nilai dari fungsi tujuan.
- b. Algoritma genetika mempunyai tujuan untuk memaksimalkan nilai *fitness* atau mencari nilai *fitness* maksimal.
- c. Untuk permasalahan minimalisasi, nilai *fitness* adalah inversi dari nilai minimal yang diharapkan. Proses inversi dapat dilakukan dengan:

$$fitness = A - F(X) \quad \text{atau} \quad fitness = \frac{A}{F(X) + \varepsilon}$$

Keterangan:

$A$  = Konstanta yang ditentukan

$X$  = Individu (kromosom)

$\varepsilon$  = Bilangan kecil yang ditentukan untuk menghindari pembagi nol atau  
 $F(X) = 0$

Kriteria yang digunakan pada proses seleksi ini adalah kriteria fungsi fitness. Masing-masing rute pada populasi awal dihitung jarak, nilai *fitness*, probabilitas *fitness* dan probabilitas kumulatif fitnessnya. Tahap-tahap perhitungan *fitness*nya adalah sebagai berikut:

1. Mencari jarak tempuh tiap rute ( $Z_i$ )
2. Mencari total jarak dari seluruh *rute* ( $\sum_{i=1}^N Z_i$ )
3. Mencari nilai *fitness* tiap *rute* ( $f_i$ )

$$f_i = \frac{\sum_{i=1}^N Z_i}{Z_i}$$

4. Mencari total *fitness* ( $\sum_{i=1}^N f_i$ )
5. Mencari probabilitas tiap *rute* ( $p_i$ )

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

6. Mencari probabilitas kumulatif tiap *rute* ( $q_i$ )

$$q_i = \sum_{k=1}^i p_i$$

Selanjutnya pemilihan sebuah *rute* yang menghasilkan populasi berikutnya dilakukan dengan cara mengambil  $N$  buah bilangan random  $r$  dengan  $0 < r < 1$  dan membandingkan bilangan random tersebut dengan probabilitas kumulatif *fitness* tiap *rute*.

### 2.3 Bilangan *biner*

Bilangan *biner* adalah bilangan yang terdiri dari 2 simbol angka 0 dan 1. Bilangan *biner* dikenal dengan bilangan basis dua, karena tiap angka biner menggunakan basis 2.

Nilai dari setiap posisi/angka biner dimulai dari ujung kanan sampai posisi terakhir disebelah kiri.

Contoh:

$$11001_2 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 25_{10}$$

Dalam kasus TSP, bilangan *biner* diasumsikan sebagai menukar gen yang dipilih secara acak dengan gen yang sesudahnya. Misalkan :

Kromosom I = [A-B-C-D-E] = [1 1 1 0 1]

Kromosom II = [B-C-A-E-D] = [1 10 0 1 ]

Akan menjadi

Kromosom I = [A-B-C-E-D] = [1 1 1 1 0]

Kromosom II = [ B-A-C-D-E] = [1 0 1 1 0]

### 2.4 Persilangan (*crossover*)

Operator persilangan merupakan operasi yang bekerja untuk menggabungkan dua kromosom orangtua (*parent*) menjadi kromosom baru (*offspring*). Tidak semua kromosom mengalami persilangan. Jumlah kromosom dalam populasi yang mengalami persilangan ditentukan oleh paramater yang disebut dengan *crossover rate* (probabilitas persilangan) .

Pindah silang pada TSP dapat diimplementasikan dengan skema *order crossover*. Kromosom yang dijadikan induk dipilih secara acak dan jumlah kromosom yang dicrossover dipengaruhi oleh parameter *crossover probability* (*pc*). Misal kita tentukan *pc* = 25%, maka diharapkan dalam 1 generasi ada 50% (3 kromosom) dari populasi mengalami *crossover*. Pertama kita bangkitkan bilangan acak R sebanyak jumlah populasi yaitu 6 kali:

Kromosom[1] = [B D E C]

Kromosom[2] = [D B E C]

Kromosom[3] = [C B D E]

Kromosom[4] = [E B C D]

Kromosom[5] = [E C B D]

Kromosom[6] = [C D E B]

Kasus TSP ini bilangan acaknya adalah antara 1-3. Misal diperoleh bilangan acaknya 1, maka gen yang ke-1 pada kromosom induk pertama diambil kemudian ditukar dengan gen pada kromosom induk kedua yang belum ada pada induk pertama dengan tetap memperhatikan urutannya. Bilangan acak untuk 3 kromosom induk yang akan di *crossover* :

C[2] = 2

C[3] = 1

C[6] = 2

Proses *crossover* :

Kromosom[2] = Kromosom[2]  $\times$  Kromosom[3]

= [B D E C]  $\times$  [C B D E]

= [B D C E]

Kromosom[3] = Kromosom[3]  $\times$  Kromosom[6]

= [C B D E]  $\times$  [C D E B]

= [C D E B]

Kromosom[6] = Kromosom[6]  $\times$  Kromosom[2]

= [C D E B]  $\times$  [B D E C]

= [C D B E]

Populasi setelah di-*crossover* :

Kromosom[1] = [D B E C]

Kromosom[2] = [B D C E]

Kromosom[3] = [C D E B]

Kromosom[4] = [E C B D]

Kromosom[5] = [E B C D]

Kromosom[6] = [C D B E]

### 2.4.1 Strategi penentuan *crossover*

Ada beberapa startegi dalam penentuan *crossover* antara lain:

- Penentuan *Crossover* harus menjamin bahwa solusi masih berada di ruang solusi.  
 $individu(k) \in S$
- Crossover* untuk kromosom *biner* dilakukan dengan menggunakan pertukaran gen antar induk.
- Crossover* untuk kromosom float dilakukan dengan pertukaran gen atau pertukaran aritmatika antar induk.
- Probabilitas *crossover* yang baik berada pada kisaran 0.5 sampai dengan 0.95.

## 2.5 Mutasi

Setelah *crossover* dilakukan, proses reproduksi dilanjutkan dengan mutasi. Hal ini dilakukan untuk menghindari solusi-solusi dalam populasi mempunyai nilai lokal optimum. Mutasi adalah proses mengubah gen dari keturunan secara random. Untuk pengkodean biner maka mutasi mengubah bit 0 menjadi bit 1 dan bit 1 menjadi bit 0.

Pada kasus TSP jumlah kromosom yang mengalami populasi ditentukan oleh parameter *mutation rate*. Proses mutasi dilakukan dengan cara menukar gen yang dipilih secara acak dengan gen yang sesudahnya. Jika gen tersebut berada diakhir kromosom, maka ditukar gen yang pertama. Pertama kita hitung dulu gen yang ada dalam populasi:

Panjang total gen= jumlah gen dalam satu kromosom \* jumlah kromosom.

Misalkan jumlah gen 4 dan jumlah kromosom ada 6. jadi  $4*6 = 24$ .

Kromosom[1] = [D B E C]

Kromosom[2] = [B D C E]

Kromosom[3] = [C D E B]

Kromosom[4] = [E C B D]

Kromosom[5] = [E B C D]

Kromosom[6] = [C D B E]

Memilih posisi gen yang mengalami mutasi dilakukan dengan membangkitkan bilangan acak 1-24. misal kita tentukan  $pm = 20\%$  maka jumlah gen yang akan dimutasi adalah  $0,2 * 24 = 4,8 = 5$ . Lima buah fungsi gen yang akan dimutasi, setelah diacak adalah posisi 3,7,10,20,24.

Proses mutasi:

Kromosom[1] = [D B C E]

Kromosom[2] = [B D E C]

Kromosom[3] = [C E D B]

Kromosom[4] = [E C B D]

Kromosom[5] = [D B C E]

### 2.5.1 Strategi menentukan mutasi

Strategi dalam menentukan nilai mutasi adalah:

- Pada kromosom *biner*, mutasi dilakukan dengan mengubah gen *biner* 0 menjadi 1 dan 1 menjadi 0.

$$a(k) = 1 - a(k)$$

- Pada kromosom *float* ada dua macam mutasi yang banyak dilakukan yaitu *random mutation* dan *shift mutation*.

- Random mutation* adalah mengganti gen yang termutasi dengan nilai acak.

$$a(k) = \text{random}$$

- Shift mutation* adalah menggeser nilai gen termutasi sebesar  $\varepsilon$ , dimana  $\varepsilon$  adalah bilangan kecil yang ditentukan.

$$a(k) = a(k) + \varepsilon$$

- Probabilitas mutasi yang baik berada pada kisaran 0 sampai dengan 0.3. Probabilitas mutasi yang terlalu kecil menyebabkan terjebak dalam optimum lokal, dan probabilitas mutasi yang terlalu besar menyebabkan *konvergensi* sulit didapatkan.

### 2.6 Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma Genetika banyak dipakai pada aplikasi

bisnis, teknik maupun pada bidang keilmuan lainnya. Algoritma ini dimulai dengan kumpulan solusi yang disebut dengan populasi. Solusi-solusi dari sebuah populasi diambil dan digunakan untuk membentuk populasi yang baru. Hal ini dimotivasi dengan harapan bahwa populasi yang baru dibentuk tersebut akan lebih baik daripada yang lama. Solusi-solusi yang dipilih untuk membentuk solusisolusi yang baru dipilih sesuai dengan *fitness* mereka masing-masing. Dalam buku dengan judul “*Adaption in Natural and Artificial System*” yang terbit pada tahun 1975, prinsip algoritma genetika diambil dari teori Darwin yaitu setiap makhluk hidup akan menurunkan satu atau beberapa karakter ke anak atau keturunannya. Di dalam proses tersebut dapat terjadi variasi yang disebabkan karena adanya mutasi, sehingga keturunan yang dihasilkan dapat mempunyai kelebihan bahkan tidak memiliki kekurangan dari orangtuanya.

Misalkan  $P(\text{generasi})$  adalah populasi dari satu generasi, maka secara sederhana algoritma genetika terdiri dari langkah-langkah:

- a. Generasi=0 (generasi awal)
- b. Inisialisasi populasi awal,  $P(\text{generasi})$ , secara acak.
- c. Evaluasi nilai *fitness* pada setiap individu dalam  $P(\text{generasi})$ .
- d. Kerjakan langkah-langkah berikut hingga generasi mencapai maksimum generasi:
  - 1) Generasi = generasi+1 (tambah generasi).
  - 2) Seleksi populasi tersebut untuk mendapatkan kandidat induk,  $P'(\text{generasi})$ .
  - 3) Lakukan *crossover* pada  $P'(\text{generasi})$ .
  - 4) Lakukan mutasi pada  $P'(\text{generasi})$ .
  - 5) Lakukan evaluasi *fitness* setiap individu pada  $P'(\text{generasi})$ .
  - 6) Bentuk populasi baru:  $P(\text{generasi}) = \{P(\text{generasi}-1) \text{ yang survive, } P'(\text{generasi})\}$



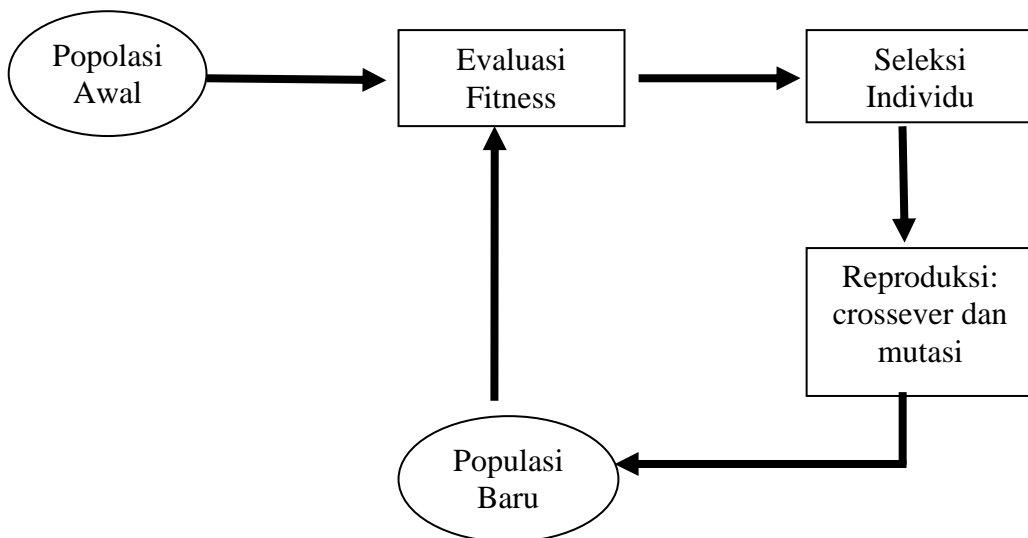
### 2.6.1 Ciri-Ciri Permasalahan yang Membutuhkan Algoritma Genetika

Adapun ciri-ciri permasalahan yang membutuhkan algoritma genetika antara lain:

- a. Mempunyai fungsi tujuan optimalisasi non linear dengan banyak kendala yang juga non linear.
- b. Mempunyai kemungkinan solusi yang jumlahnya tak berhingga.
- c. Membutuhkan solusi “*real-time*” dalam arti solusi bisa didapatkan dengan cepat sehingga dapat diimplementasi untuk permasalahan yang mempunyai perubahan yang cepat seperti optimasi pada pembebanan kanal pada komunikasi seluler.
- d. Mempunyai *multi objective* dan *multi criteria*, sehingga diperlukan solusi yang dapat secara bijak diterima oleh semua pihak.

### 2.6.2 Siklus algoritma genetika

Siklus dari algoritma genetika pertama kali dikenalkan oleh David Goldberg, dimana gambar siklus tersebut. Dapat dilihat pada gambar 2.2 berikut:



**Gambar 2.2** Siklus Algoritma Genetika oleh David Goldberg

## 2.7 Individu algoritma genetika

Individu merupakan kumpulan gen dalam suatu algoritma genetika bisa dikatakan sama dengan kromosom, gen ini bisa biner, float, dan kombinatorial. Individu dalam algoritma genetika dapat juga menyatakan salah satu kemungkinan solusi yang dicari. Misalkan dalam traveling salesman problem individu dapat menyatakan suatu jalur terpendek yang akan ditempuh.

### 2.7.1 Strategi menentukan individu

Strategi dalam menentukan individu pada algoritma genetika adalah:

- a. Individu adalah suatu komposisi nilai yang menyatakan solusi dari suatu permasalahan.
- b. Permasalahan optimasi adalah suatu permasalahan yang mempunyai banyak solusi dan harus bisa ditentukan solusi mana yang dikatakan optimal.
- c. Berdasarkan definisi individu algoritma genetika dibedakan menjadi 2 macam yaitu *static genetic algorithm* (ukuran individu sama untuk satu populasi) dan *dynamic genetic algorithm* (ukuran individu tidak sama untuk satu populasi).

### 2.7.2 Definisi Individu Untuk Menentukan Nilai Maksimal Fungsi $F(x,y,z)$

Definisi individu untuk menentukan nilai maksimum fungsi  $F(x,y,z)$  ada 2 yaitu:

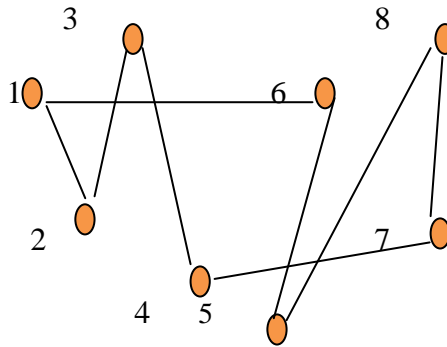
- a. Nilai fungsi  $F(x,y,z)$  sangat tergantung pada nilai  $(x,y,z)$ . Artinya solusi yang dicari adalah nilai  $(x,y,z)$  yang menyebabkan nilai  $F(x,y,z)$  maksimum.
- b. Individu didefinisikan sebagai nilai  $(x,y,z)$  dan dituliskan dengan:

$$S = \{(x, y, z) \mid x, y, z \in R\}$$

### 2.7.3 Definisi Individu Untuk TSP

Definisi individu pada kasus *travelling salesman problem* yaitu:

- TSP (*Traveling Salesman Problem*) adalah suatu permasalahan dimana seorang sales harus mengunjungi N kota dengan jarak yang paling pendek, dengan syarat satu kota hanya dikunjungi satu kali.
- Solusi TSP adalah jalur yang melewati semua kota dan jaraknya paling pendek.
- Individu untuk TSP didefinisikan sebagai jalur atau urutan nomor kota yang dikunjungi. Misalkan untuk 8 kota salah satu jalur yang mungkin adalah : 1-2-3-4-7-8-5-6



Gambar 2.3 Individu pada *Travelling Salesman Problem*

### BAB III

#### METODOLOGI PENELITIAN

Studi literature dalam menyelesaikan penelitian ini yaitu dengan langkah-langkah sebagai berikut:

- a. Memahami algoritma genetika *multi objektif*.

Dalam penelitian skripsi ini akan dijelaskan penjelasan tentang algoritma genetika *multi obyektif* dan bagian-bagian dari algoritma genetika *multi obyektif*.

- b. Memahami bentuk permasalahan atau kendala yang dihadapi dalam mencari jalur terpendek menggunakan algoritma genetika *multi objektif*.

Dengan algoritma genetika *multi obyektif* akan ditentukan jarak terpendek, serta akan dicari waktu dan biaya paling murah.

- c. Menyelesaikan permasalahan *travelling salesman problem* dengan menggunakan algoritma genetika *multi objektif*.

Penggunaan bahasa pemograman *bornland Delphi* untuk membantu menyelesaikan permasalahan *traveling salesman problem*.

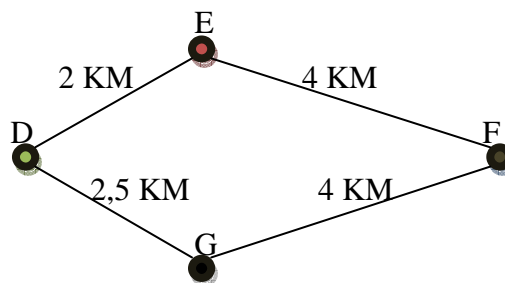
## BAB IV

### APLIKASI ALGORITMA GENETIKA *MULTI OBYEKTIF* PADA *TRAVELLING SALESMAN PROBLEM*

Pembahasan pada bab ini berkaitan dengan penyelesaian aplikasi algoritma genetika *multi obyektif* pada *travelling salesman problem* dan beberapa teori pendukung penyelesaian kasus perjalanan *mini bus* sukajadi-panam. Yaitu:

#### 4.1 *Travelling Salesman Problem (TSP)*

Persoalan TSP melibatkan seorang salesman (penjaja dagangan) yang harus melakukan kunjungan pada setiap kota yang menjadi bagiannya untuk menjajakan produknya. Rangkaian kota-kota yang dia kunjungi dinamakan lintasan, dimana dalam lintasan tersebut terdapat batasan yaitu tidak boleh ada lebih dari satu kota yang sama. Dengan kata lain, dalam mengunjungi kota-kota, penjaja tidak boleh singgah pada suatu kota lebih dari satu kali. Dan pada akhir perjalanan dia harus kembali ke kota tempat dia memulai perjalanan. Dapat dilihat pada contoh gambar berikut:



Gambar 4.1 contoh kasus *Tavelling Salesman Problem*

Berdasarkan gambar diatas terdapat empat kota, yaitu D, E, F, dan G. Lintasan yang ditempuhnya adalah dari kota E menuju ke kota D menuju ke kota G kemudian menuju ke kota F. Setelah sampai di kota F, maka dia harus kembali ke kota E. Penyelesaian dari persoalan ini adalah nilai optimum dari rute yang paling murah, yaitu perjalanan dengan jarak terpendek atau yang mempunyai total harga minimum.

TSP juga merupakan salah satu bagian dari persoalan transportasi. Pada persoalan transportasi selain masalah sarana transportasi, efisiensi pengiriman suatu barang juga ditentukan oleh lintasan yang diambil untuk mengirimkan barang tersebut. Oleh karena itu solusi optimal dari permasalahan TSP akan sangat membantu perusahaan dalam pengiriman barang untuk mengefesiansikan proses pengiriman, baik dari segi waktu dan biaya.

Berdasarkan definisi TSP ini, dapat kita simpulkan bahwa terdapat dua batasan dalam persoalan ini. Batasan pertama adalah bahwa setiap kota tidak boleh dikunjungi lebih dari satu kali dan pada akhir perjalanan harus kembali ke kota asal, dengan kata lain lintasan berbentuk siklik. Batasan yang kedua bahwa lintasan yang ditempuh adalah lintasan yang paling minimum atau terpendek. Data yang diketahui dalam permasalahan ini adalah jumlah kota yang harus dikunjungi beserta jarak antar kotanya.

#### **4.2 TSP *Multi objective***

*Traveling Salesman Problem Multi objective* merupakan pengembangan dari permasalahan TSP, dimana sasaran yang diterapkan pada permasalahan TSP bukan lagi terfokus pada lintasan dengan jarak yang paling pendek, tetapi dapat diterapkan sasaran lain yang sesuai dengan plan atau kasus yang akan diselesaikan. Dalam penelitian ini kasus yang diterapkan adalah tentang perjalanan mini bus (oplet) melalui terminal-terminal dalam kota Pekanbaru. Jadi dalam kasus ini seorang traveling salesman diibaratkan sebagai sebuah *mini bus*, sedangkan kota yang dilalui adalah terminal-terminal itu sendiri, selebihnya akan sama dengan TSP yang telah dijelaskan sebelumnya. Namun ada perbedaan mengenai sasaran yang diterapkan yaitu untuk mencari lintasan dengan waktu tempuh yang paling pendek serta biaya yang paling sedikit. Data yang digunakan pada penelitian ini adalah data lintasan yang dilalui setiap mini bus di kota Pekanbaru. Oleh karena itu sangat sulit menentukan titik-titik selain terminal yang dilewati oleh mini bus karena harus melalui survey khusus. Karena itu pada penelitian ini dibatasi titik-titik yang dilalui oleh mini bus adalah terminal dalam kota Pekanbaru.

Penyelesaian TSP *multi obyektif* dengan algoritma genetika akan berdampak pada fungsi *fitness* yang diterapkan untuk kasus tersebut, sekaligus yaitu waktu dan biaya. Untuk itu diperlukan suatu perangkat lunak dalam menyelesaikan penelitian skripsi ini, maka penulis menggunakan bahasa pemrograman *borland delphi 7.0*.

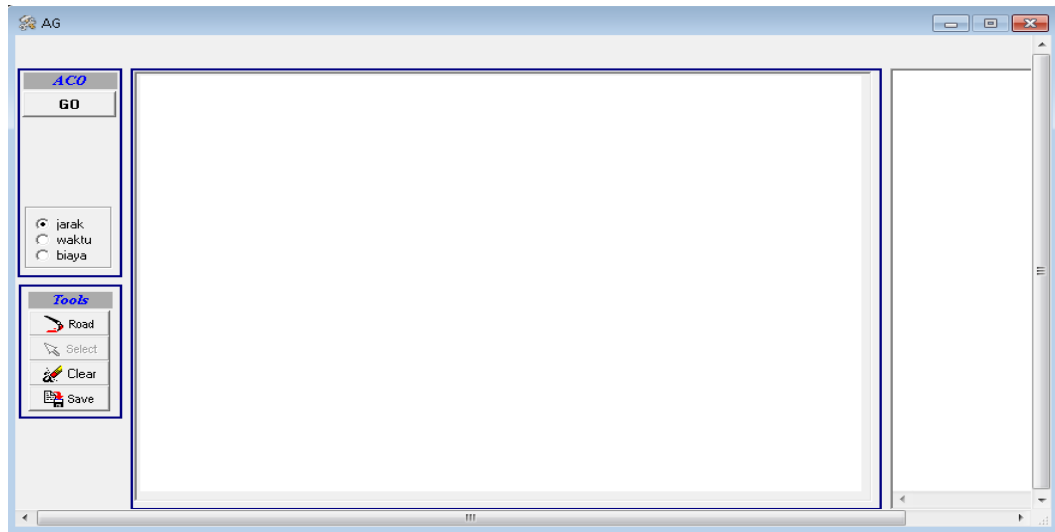
#### **4.2.1 Bahasa Pemrograman yang digunakan Borland Delphi 7.0.**

Bahasa pemrograman yang digunakan adalah *borland delphi 7.0*, Pemilihan perangkat lunak didasarkan pertimbangan sebagai berikut:

1. Merupakan *event-driven programming* (pemrograman terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa *event* atau kejadian tertentu (tombol diklik, menu dipilih, dan lain-lain).
2. Mempunyai kemampuan untuk memanfaatkan *windows* dan tidak memerlukan pemrograman khusus untuk menampilkan jendela (*window*), dan dapat menampilkan simpul-simpul kota yang akan dilalui sebuah mini bus.
3. Dapat digunakan untuk menguji program (*debugging*) dan menghasilkan program akhir EXE, yang bersifat *executable* atau dapat langsung dijalankan.

#### **4.3 Tampilan aplikasi**

Aplikasi ini dirancang khusus untuk membantu dalam pencarian kasus penelitian skripsi ini. Dengan menggunakan aplikasi ini akan mudah dalam mencari permasalahan *travelling salesman problem*. Tampilan aplikasi Borland delphi 7.0 dapat lihat pada gambar 4.2 sebagai berikut:



Gambar 4.2 Tampilan Menu Utama Algoritma Genetika *Multi Obyektif*

Penggunaan aplikasi ini akan dapat menentukan titik-titik kota yang akan dilalui *mini bus* sukajadi-panam.

#### 4.4 Contoh Aplikasi

Aplikasi algoritma genetika *multi obyektif* pada *travelling salesman problem* dengan studi kasus perjalanan *mini bus* jalur sukajadi-panam. Melalui survei langsung diperoleh jarak antar jalur kota sebagai berikut:

Tabel 4.3 Jarak Antar Kota yang akan dilalui Mini Bus Sukajadi-Panam

NO	Jalur Kota	Jarak Lurus (Meter)
1	JL.TUANKU TAMBUSAI I (Plaza Matahari-Lampu rambu lalu lintas Mol SKA)	1260 M
2	JL.TUANKU TAMBUSAI II (Lampu rambu Lalu lintas mol SKA-Bundaran Arengka II)	1350 M
3	TERMINAL PAYUNG SEKAKI ( AKAP atau Jalan baru)	3040 M
4	JL.SOEKARNO-HATTA (Lampu rambu lalu lintas mol SKA-Lampu merah pasar pagi arengka I)	2820 M
5	Jl.M.YAMIN (bundaran Arengka II-Lampu	2550 M



	Rambu Lalu Lintas Panam)	
6	JL.H.R.SUBRANTAS I (Lampu merah pasar pagi arengka I- Lampu Rambu Lalu Lintas Panam)	1770 M
7	JL.H.R.SUBRANTAS II(Lampu Rambu Lalu Lintas Panam-simpang empat pekanbaru dan bangkinang)	1830 M
8	JL.GARUDA SAKTI I (simpang empat pekanbaru bangkinang-simpang buluh cina)	450 M
9	JL.GARUDA SAKTI II ( jalan baru AKAP-simpang buluh cina)	1950 M
10	JL.BULUH CINA( simpang buluh cina-UIN)	360 M

Misalkan 3 orang supir mini bus (oplet) A, B, C dan tiga buah mini bus (oplet) nya yang akan membawa penumpang dari sukajadi kekampus uin susqa riau, isi dari penumpang oplet berbeda-beda. Mini bus A membawa penumpang 1 orang mahasiswa universitas islam negeri (UIN) dan 7 orang masyarakat awam, mini bus B membawa penumpang sama dengan mini bus A yaitu 1 orang mahasiswa uin dan 7 orang masyarakat awam, sedangkan mini bus C membawa 1 orang mahasiswa uin dan 6 orang masyarakat awam. jalur yang akan dilalui mini bus itu ada 3 yaitu:

1. Jalur Arengka I

Jalur Arengka I yaitu: T.Tambusai I-Soekarno Hatta-H.R. Subrantas I-H.R.Subrantas II-Garuda Sakti I-Buluh Cina-Uin.

2. Jalur Arengka II

Jalur Arengka II yaitu: T.Tambusai I-T.Tambusai II-M.Yamin-H.R.Subrantas II-Garuda Sakti-Buluh Cina-Uin.

3. Jalur AKAP (Antar Kota Antar Propinsi) atau Terminal Payung Sekaki

Jalur Terminal Payung Sekaki yaitu: T.Tambusai I-T.Tambusai II-Akap atau Terminal Payung Sekaki-Garuda Sakti II-Buluh Cina-Uin.

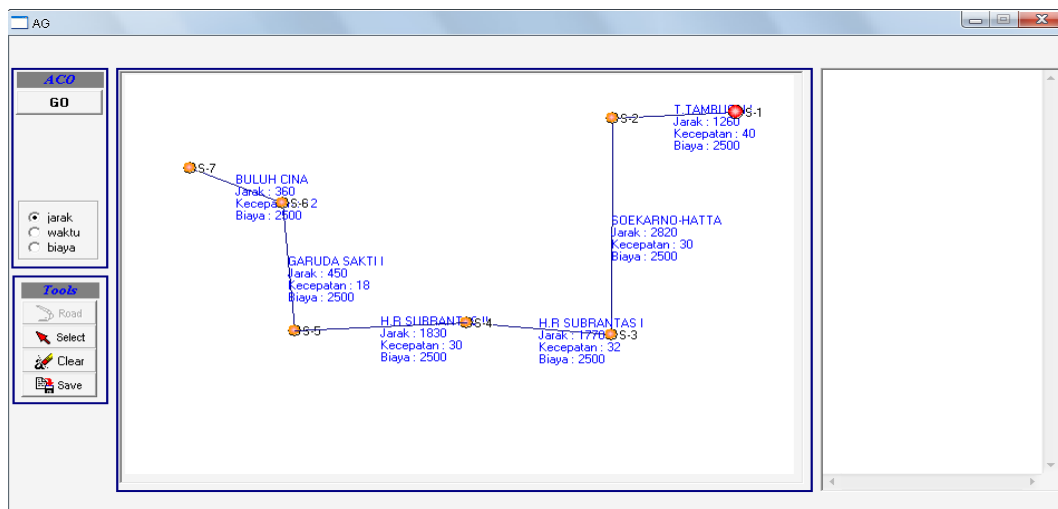
Berdasarkan survei langsung yang dilakukan secara berulang sebanyak 3 kali dapat diperoleh bahwa kecepatan antar kota yang akan dilalui mini bus sukajadi-panam berbeda-beda. Sebagai berikut:

Tabel 4.4 Kecepatan mini bus yang melewati jalur Sukajadi-Panam

NO	Jalur Kota	Kecepatan (Km/Jam)
1	JL.TUANKU TAMBUSAI I (Plaza Matahari-Lampu rambu lalu lintas Mol SKA)	40 Km/Jam
2	JL.TUANKU TAMBUSAI II (Lampu rambu Lalu lintas mol SKA-Bundaran Arengka II)	52 Km/Jam
3	TERMINAL PAYUNG SEKAKI ( AKAP atau Jalan baru)	78 Km/Jam
4	JL.SOEKARNO-HATTA (Lampu rambu lalu lintas mol SKA-Lampu merah pasar pagi arengka I)	30 Km/Jam
5	Jl.M.YAMIN (bundaran Arengka II-Lampu Rambu Lalu Lintas Panam)	40 Km/Jam
6	JL.H.R.SUBRANTAS I (Lampu merah pasar pagi arengka I- Lampu Rambu Lalu Lintas Panam)	32 Km/Jam
7	JL.H.R.SUBRANTAS II(Lampu Rambu Lalu Lintas Panam-simpang empat pekanbaru dan bangkinang)	30 Km/Jam
8	JL.GARUDA SAKTI I (simpang empat pekanbaru bangkinang-simpang buluh cina)	18 Km/Jam
9	JL.GARUDA SAKTI II ( jalan baru AKAP-simpang buluh cina)	50 Km/Jam
10	JL.BULUH CINA( simpang buluh cina-UIN)	12 Km/Jam

Berdasarkan hasil survey langsung diperoleh bahwa biaya perjalanan mini bus dari titik jalur yang akan disinggahi adalah sama yaitu Rp 2.500,00

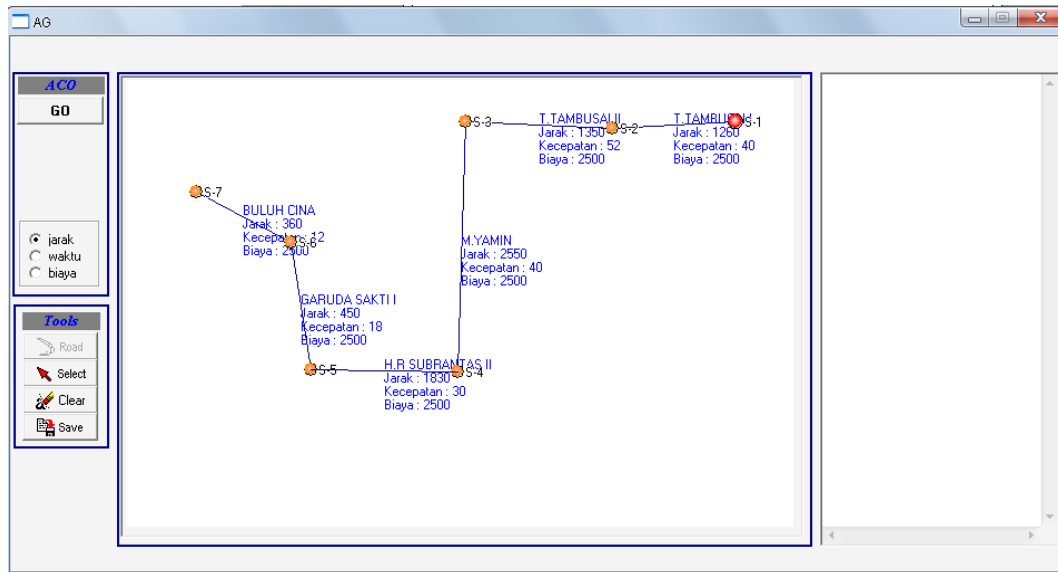
Berdasarkan ketiga jalur tersebut akan ditentukan jarak terpendek, waktu terpendek atau waktu tercepat dan biaya yang didapat seorang supir mini bus .Jalur yang akan dilalui masing-masing supir mini bus berbeda-beda. Untuk mini bus A akan melalui jalur 1 yaitu arengka I karena masyarakat awam ada yang akan turun pada jalur 1. Simpul atau titik-titik kota dilambangkan dengan S. Dapat dilihat pada gambar 4.5 berikut:



Gambar 4.5 Jalur Arengka I

Berdasarkan gambar 4.5, dapat dilihat bahwa kromosom pada jalur yang dilalui oleh supir mini bus A adalah T.Tambusai-Soekarno Hatta-H.R.Subrantas I-H.R.Subrantas II-Garuda Sakti I-Buluh Cina-Uin.

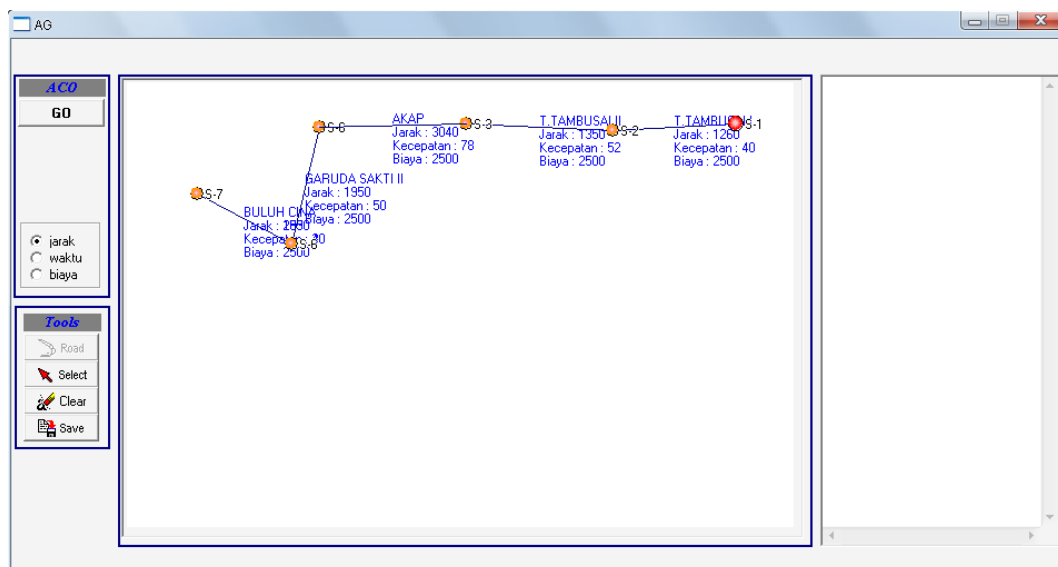
Mini bus B akan melalui jalur arengka II karena adanya penumpang masyarakat awam yang akan turun pada jalur 2. dapat dilihat pada gambar 4.6 berikut:



Gambar 4.6 Jalur Arengka II

Berdasarkan gambar 4.6 dapat dilihat bahwa kromosom pada jalur yang dilalui oleh supir mini bus B adalah T.Tambusai I-T.Tambusai II-M.Yamin-H.R.Subrantas II-Garuda Sakti I-Buluh Cina-Uin.

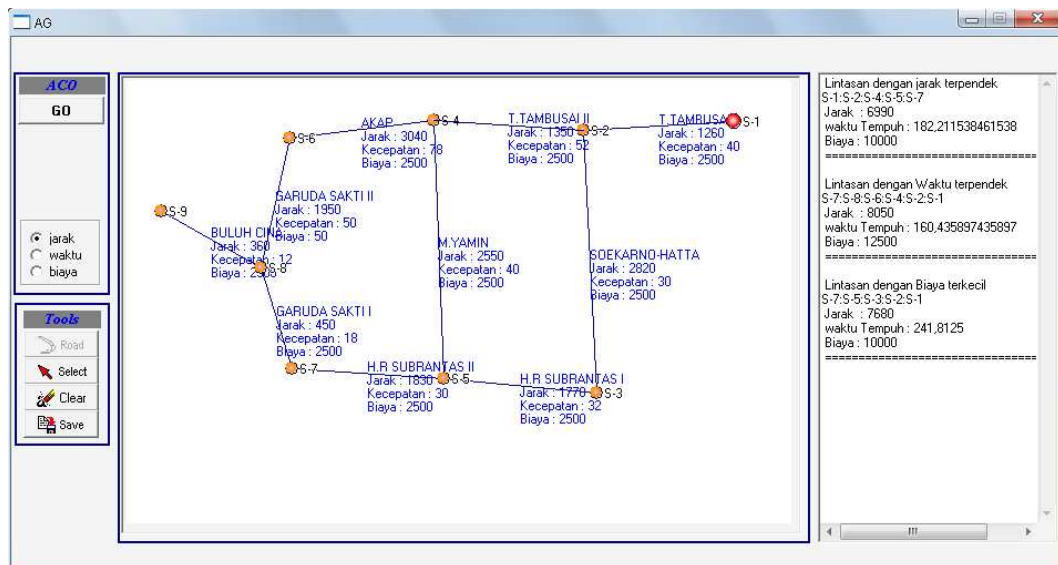
Mini bus C akan melalui jalur akap atau terminal payung sekaki karena adanya penumpang awam yang akan turun pada jalur 3. dapat dilihat pada gambar 4.7 berikut:



Gambar 4.7 Jalur Akap atau Terminal Payung Sekaki

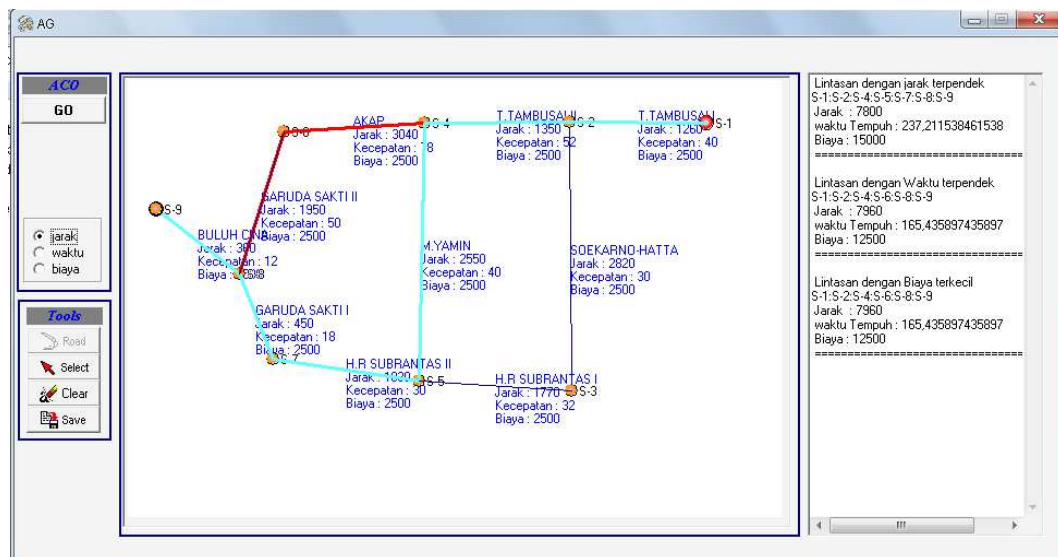
Berdasarkan gambar 4.7 dapat dilihat kromosom pada jalur yang dilalui oleh supir mini bus C adalah T.Tambusai I-T.Tambusai II-Akap-Garuda Sakti II-Buluh Cina-Uin.

Jalur-jalur yang dilalui mini bus itu dapat digabungkan semua jalur yang dilalui tiga mini bus tersebut. Dapat dilihat pada gambar 4.8 berikut:



Gambar 4.8 Gabungan jalur-jalur yang dilalui mini bus

Berdasarkan dari gabungan jalur-jalur yang dilalui mini bus tersebut akan dapat ditentukan jarak terpendek, waktu terkecil dan biaya paling sedikit. Dapat dilihat pada gambar 4.9 berikut:



Gambar 4.9 jalur terpendek, waktu terkecil dan biaya terkecil

Kasus *travelling salesman problem* ini dapat disimpulkan bahwa jarak terpendek, waktu terkecil dan biaya paling sedikit dapat diselesaikan dengan bahasa pemrograman *borland delphi* serta mendapatkan hasil yang baik. Diperlihatkan hasil sebagai berikut:

- Lintasan dengan jarak terpendek  
S-1:S-2:S-4:S-5:S-7:S-8:S-9  
Jarak : 7800  
waktu Tempuh : 237,211538461538  
Biaya : 15000
- Lintasan dengan Waktu terpendek  
S-1:S-2:S-4:S-6:S-8:S-9  
Jarak : 7960  
waktu Tempuh : 165,435897435897  
Biaya : 12500
- Lintasan dengan Biaya terkecil  
S-1:S-2:S-4:S-6:S-8:S-9  
Jarak : 7960  
waktu Tempuh : 165,435897435897  
Biaya : 12500

Lintasan TSP tersebut dapat dijelaskan bahwa dalam pencarian jarak terpendek dengan kriteria:

Kromosom I : S1-S2-S3-S5-S7-S8-S9

$$1260-2820-1770-1830-450-360 = 8490 \text{ Meter}$$

Kromosom II : S1-S2-S4-S5-S7-S8-S9

$$1260-1350-2550-1830-450-360 = 7800 \text{ Meter}$$

Kromosom III : S1-S2-S4-S6-S8-S9

$$1260-1350-3040-1950-360 = 7960 \text{ Meter}$$

Jadi dapat disimpulkan bahwa fungsi fitness yang terbaik terdapat pada Kromosom II, karena mempunyai jarak paling pendek yaitu 7800 Meter dibandingkan Kromosom lainnya.

Untuk pencarian waktu terkecil dalam kasus TSP *multi obyektif* ini ditentukan berdasarkan  $\frac{\text{jarak}}{\text{kecepatan}}$  ketiga mini bus dalam melewati jalur-jalur

yang sudah ditentukan, dapat dicari dengan kriteria:

Kromosom I : S1-S2-S3-S5-S7-S8-S9

$$31,5-94-55,31-61-25-30 = 296,81 \text{ Menit}$$

Kromosom II : S1-S2-S4-S5-S7-S8-S9

$$31,5-25,96-63,75-61-25-30 = 237,21 \text{ Menit}$$

Kromosom III : S1-S2-S4-S6-S8-S9

$$31,5-25,96-38,97-39-30 = 165,43 \text{ Menit}$$

Jadi dapat disimpulkan bahwa dalam pencarian waktu terbaik dengan rumus  $\frac{\text{jarak}}{\text{kecepatan}}$ , maka dapat ditentukan bahwa kromosom III adalah waktu terkecil dengan 165,43 Menit dibandingkan dengan kromosom lainnya.

Masalah pencarian biaya paling sedikit yang akan diperoleh oleh supir mini bus itu, dapat ditentukan dengan kriteria:

Kromosom I : S1-S2-S3-S5-S7-S8-S9

$$2500-2500-2500-2500-2500-2500 = 15000$$

Kromosom II : S1-S2-S4-S5-S7-S8-S9

$$2500-2500-2500-2500-2500-2500 = 15000$$

Kromosom III : S1-S2-S4-S6-S8-S9

$$2500-2500-2500-2500-2500 = 12500$$

Jadi dapat disimpulkan bahwa biaya paling sedikit yang didapat supir mini bus itu pada kromosom III adalah Rp 12500,00.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan pembahasan pada Bab IV, penyelesaian Aplikasi Algoritma Genetika Multi Obyektif pada *Traveling Salesman Problem* , dapat diambil kesimpulan sebagai berikut:

1. *Travelling Saleman Problem* dapat diselesaikan dengan algoritma genetika multi obyektif.
2. Jarak terpendek, waktu terkecil dan biaya paling sedikit pada kasus perjalanan mini bus Sukajadi Panam dapat ditentukan dengan pemograman *Borland Delphi 7.0*, yaitu:
  - a. Lintasan dengan jarak terpendek pada kromosom II yaitu T.Tambusai I-T.Tambusai II-M.Yamin-H.R.Subrantas II-Garuda Sakti-Buluh Cina-Uin dengan jarak 7800 meter.
  - b. Lintasan dengan waktu terpendek pada kromosom III yaitu T.Tambusai I-T.Tambusai II-Akap atau Terminal Payung Sekaki-Garuda Sakti II-Buluh Cina-Uin dengan waktu 165,43 Menit.
  - c. Lintasan dengan biaya terkecil pada kromosom III yaitu T.Tambusai I-T.Tambusai II-Akap atau Terminal Payung Sekaki-Garuda Sakti II-Buluh Cina-Uin dengan biaya Rp12.500,00.

#### **5.2 Saran**

Beberapa hal yang disarankan dalam pengembangan aplikasi *algoritma multi obyektif* pada *traveling salesman problem* sebagai berikut:

1. *Traveling salesman problem* pada penelitian diselesaikan dengan aplikasi algoritma genetika *multi obyektif*, tidak menutup kemungkinan kasus TSP ini dapat diselesaikan dengan aplikasi lainnya misalnya algoritma *heuristic* dan graf berbobot.

2. Pengembangan penggunaan algoritma genetika *multi obyektif* dapat menggunakan pemograman selain *Borland Delphi 7.0*. misalnya pencarian secara manual, dan pemograman yang mendukung terbentuknya simpul-simpul pada kasus *traveling salesman problem*.

## DAFTAR PUSTAKA

- Basuki Achmad. "*Starategi menggunakan Algoritma genetika*", Politeknik Elektronika Negeri Surabaya. PENS-ITS.2003.
- Basuki Achmad. "*Tips dan Trik : Algoritma Genetika*", Politeknik Elektronika Negeri Surabaya. PENS-ITS.2006.
- Hermawanto Denny. " <http://soft-computing.org,2010>"URL: diakses pada tanggal 02 januari 2010.
- Kusumadewi Sri. "*Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*", Graha Ilmu.Yogyakarta.2005.
- Wikipedia. " [IF14058@students.if.itb.ac.id,2010](http://IF14058@students.if.itb.ac.id,2010)"URL: [diakses](#) pada tanggal 02 januari 2010.
- Zuhaimi. "*Pembangunan Kaedah Heuristik Berasaskan Algoritma Genetik Untuk Menylesaikan Masalah Perjalanan Kendaraan*", Skripsi Jabatan Matematik Fakultas Sains Universiti Teknologi Malaysia, 2008.